AMENDMENTS TO THE SPECIFICATION

The following amendments to the Specification are submitted as variously required by the Examiner and to improve consistency with the Drawing Figures.

No new matter is involved with any Specification Amendment.

Please replace paragraph [0012] of the Specification with the following replacement paragraph:

[0012] FIG. 4 illustrates in flowchart form the process executed by the translational ~~look-ahead~~ look-aside buffer (TLB) in accordance with a preferred embodiment.

Please replace paragraph [0013] of the Specification with the following replacement paragraph:

[0013] Referring now to FIG. 1, a block diagram representation of a multi-thread processor is shown with an accompanying translational ~~look-up~~ look-aside buffer (TLB) in accordance with a preferred embodiment of the invention. The multi-thread CPU computing system 10 is illustrated as multiple processors 11A-11N, representing N threads of a multi-thread CPU. Each of the threads 11A-11N execute in a pipeline processor. During simultaneous execution of the various threads, access to a main memory 17 may be required to complete executing an instruction.

Please replace paragraph [0014] of the Specification with the following replacement paragraph:

[0014] The access to the main memory 17 is through a memory management unit 13. Associated with the memory management unit 13 is a translational look-aside buffer (TLB) 15. The process of retrieving and writing data to the memory 17 through bus 14 is aided with the memory management unit 13 translational look-up buffer (TLB) 15. There is one instance of

physical registers to hold each of the executing threads architectural contexts. For each thread, that may be simultaneously executed, there is a copy of the ~~GPRS, the LR, the CTR, the XER and the CR~~ General Purpose Registers (GPRs), the Link Register (LR), the Count Register (CTR), the Fixed Point Exception Register (XER), and the Condition Register (CR). Each thread being processed is allocated some dedicated physical storage elements in the main memory 17 to hold the state associated with the thread represented by the contents of the architecturally defined registers. Each instruction that is in the CPU pipeline is tagged with a thread ID so that the architectural results that it produces can be applied to the correct thread's architectural resources. Thread ID register 12 maintains the thread identification ID so that the result of execution can be identified with a particular thread.

Please replace paragraph [0017] of the Specification with the following replacement paragraph:

[0017] The organization of the translational look-aside buffer (TLB) 15 in accordance with a preferred embodiment is illustrated more particularly with respect to FIG. 2. The ~~process identifier (PIED)~~ process identifier (PID) and effective address (EA) are stored as a virtual address along with a real address 26 for main memory 17 in a content addressable memory 22. The virtual address comprising process identifier (PID) and the effective address (EA) are used as the look-up keys. The translational look-aside buffer ~~22~~ 15 includes a search engine 25 which uses a portion of the virtual address comprising the process identifier (PID) and the higher order bits of the effective address (EA) to locate a particular real address (RA) stored in a location 23. FIG. 2, unlike the conventional translational look-up buffer architecture, includes a thread, implicit identifier bit (Tbit). The Tbit is used to identify whether or not the entry 23 in the TLB is associated with one of the multiple threads 11A-11N being executed.

Please replace paragraph [0019] of the Specification with the following replacement paragraph:

[0019] The virtual address (VA) which is called for by the executing thread has bits 0-39 which comprise (as shown in FIG. 3) the process identification number (PID 0:7) bits the higher order bits of the effective address (EA 0:21) bits, and the lower order (EA 22:31) bits. In accordance with the present embodiment of the invention, the virtual page number (VPN) comprises the process identification number (PID 0;7) and the higher order bits of the effective address (EA 0:21). Using the VPN (0:29), the ~~RPN,~~ real page number (RPN), representing the higher order bits of the real address RA (0:22) are located in the memory location 23.